What is claimed is:

**CLAIMS:**

What is claimed is:

1. A method, comprising:

automatically inserting into instructions of a first thread at least one instruction that relinquishes control of a multi-tasking processor to another thread that will be concurrently sharing the processor.

2. The method of claim 1, further comprising:

automatically inserting into instructions of a second thread at least one instruction that relinquishes control of the multi-tasking processor to another thread that will be concurrently sharing the processor.

3. The method of claim 2, wherein

automatically inserting into instructions of the first thread comprises inserting based on at least one characteristic of the instructions of the second thread; and

automatically inserting into instructions of the second thread comprises inserting based on at least one characteristic of the instructions of the first thread.

4. The method of claim 2, further comprising:

repeating a procedure that determines one or more locations to automatically insert instructions that relinquish control of the processor into the instructions of the first and second threads.

5. The method of claim 3,

wherein the at least one characteristic of the instructions of the first thread comprises an average number of consecutive instructions that do not relinquish control of the processor.

6. The method of claim 5,

wherein the at least one characteristic of the instructions of the first thread comprises a standard deviation derived from the number of consecutive instructions that do not relinquish control of the processor.

7. The method of claim 1, further comprising:

constructing a data flow graph of the instructions of the first thread, the data flow graph comprising an organization of nodes associated with subsets of the instructions of the first thread; and

determining at least one of the following:

a number of consecutive instructions ending a one of the nodes that do not relinquish control of the processor;

a number of consecutive instructions beginning a one of the nodes that do not relinquish control of the processor; and a number of consecutive instructions between instructions of one of the nodes that relinquish control of the processor.

8. The method of claim 1, wherein automatically inserting comprises inserting to keep intact a group of instructions identified as indivisible.

9. The method of claim 1, wherein the processor comprises a multi-threaded central processor unit (CPU).

10. The method of claim 1, wherein the processor comprises a multi-threaded engine of a multi-engine processor.

11. The method of claim 10, wherein the multi-threaded engine of the multi-engine processor comprises an engine not having any floating point instructions in the engine's instruction set.

12. A computer program product, disposed on a computer readable medium, the program including instructions to:

access instructions of a first thread; and

insert into the instructions of a first thread at least one instruction that relinquishes control of a multi-tasking processor to another thread that will be concurrently sharing the processor.

13. The program of claim 12, further comprising instructions to:

insert into instructions of a second thread at least one instruction that relinquishes control of the processor.

14. The program of claim 13, wherein the instructions to:

insert into instructions of the first thread comprises inserting based on at least one characteristic of the instructions of the second thread; and

insert into instructions of the second thread comprises inserting based on at least one characteristic of the instructions of the first thread.

15. The program of claim 13, further comprising instructions to:

repeat a procedure that determines one or more locations to automatically insert instructions that relinquish control of the processor into the instructions of the first and second threads.

16. The program of claim 14,

wherein the at least one characteristic of the instructions of the first thread comprises an average number of consecutive instructions that do not relinquish control of the processor.

17. The program of claim 16,

wherein the at least one characteristic of the instructions of the first thread comprises a standard deviation derived from the number of consecutive instructions that do not relinquish control of the processor.

18. The program of claim 1, further comprising instructions to:

construct a data flow graph of the instructions of the first thread, the data flow graph comprising an organization of nodes associated with subsets of the instructions of the first thread; and

determine at least one of the following:

a number of consecutive instructions ending a one of the nodes that do not relinquish control of the processor;

a number of consecutive instructions beginning a one of the nodes that do not relinquish control of the processor; and

a number of consecutive instructions between instructions of one of the nodes that relinquish control of the processor.

19. The program of claim 12, wherein the instructions to insert comprise instructions to insert to keep intact a group of instructions identified as indivisible.

20. The program of claim 12, wherein the processor comprises a multi-threaded central processor unit (CPU).

21. The program of claim 12, wherein the processor comprises a multi-threaded engine of a multi-engine processor.

22. The program of claim 21, wherein the multi-threaded engine of the multi-engine processor comprises an engine not having any floating point instructions in the engine's instruction set.

23. The program of claim 22, wherein the program comprises at least one of the following: a compiler, an assembler, and a source code pre-processor.

24. A method comprising:

managing execution control of a multi-tasking processor shared by multiple threads by automatically inserting instructions into at least some of the multiple threads to relinquish control of the multi-tasking processor to a different thread.

25. The method of claim 24, wherein managing comprises inserting instructions into the threads to provide a more equal distribution of processor execution control among at least some of the threads than before the inserting.

26. The method of claim 24, wherein managing comprises inserting instructions into the threads to provide a subset of the multiple threads a greater share of processor execution control than before the inserting.

27. The method of claim 24, wherein the inserting comprises inserting based on data flow graphs generated for the, respective, threads.

28. The method of claim 24, wherein the multi-tasking processor comprises a co-operative multi-tasking processor.

29. The method of claim 24, wherein the multi-tasking processor comprises a one of a set of multi-tasking processors integrated on the same semiconductor chip.